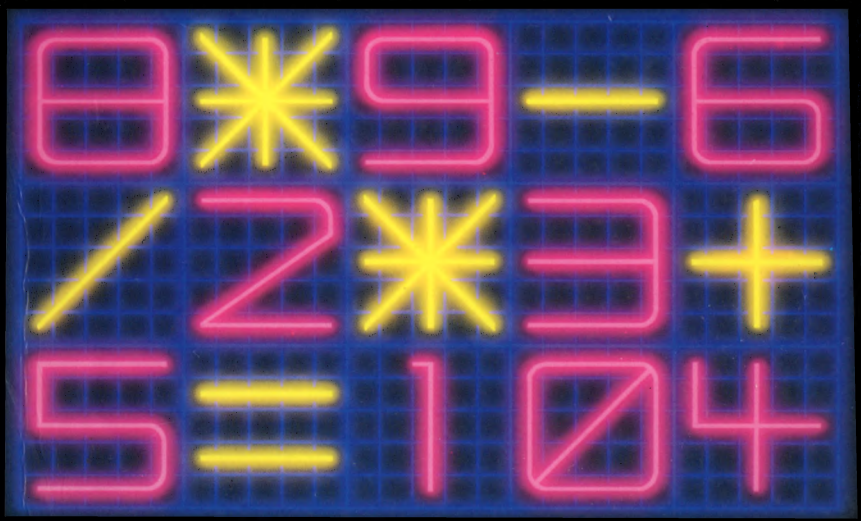
MICRO MATES

Simple Maths, Tables and Numbers



PROGRAMS FOR THE BBC MICRO AND ELECTRON COMPUTERS

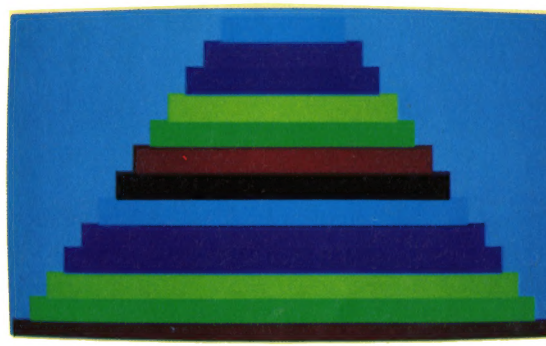
JONATHAN INGLIS

JONATHAN INGLIS

Simple Maths, Tables and Numbers

Photography
MARK GATEHOUSE

Consultant Editor
ROB PORTER



GRANADA

Published by Dragon Hardbacks 1984
Granada Publishing Limited
8 Grafton Street, London W1X 3LA

Copyright © Jonathan Inglis
Illustrations copyright © Granada Publishing

British Library Cataloguing in Publication Data
Inglis, Jonathan
Simple maths, tables and numbers. – (Micro mates; 1)
1. Mathematics – Data processing – Juvenile literature 2. BBC Microcomputer-Programming 3. Electron Microcomputer-Programming
I. Title
510'.028'5424 QA76.95

ISBN 0-246-12268-4

Printed in Italy by
New Interlitho, Milan

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the publishers.

Simple Maths, Tables and Numbers

This book is part of the MICRO MATES series. It has been written to help you use your BBC Micro or Electron microcomputer. In it you will find a whole series of programs that you can type in and try out for yourself.

MICRO MATES programs are simple, straightforward and fun to use. Every program is clearly described and explained so that you can understand what it does and why it works. The MICRO MATES series helps and encourages you to start writing programs of your own.

Along with the program descriptions you will find lots of ideas: how to change your programs, how to make them run more efficiently and how to use all the short cuts and special functions your computer has to offer.

Every young computer fan will find the MICRO MATES series full of ideas and challenges.

This book will show you how you can use your BBC Micro or Electron to do sums, work out problems and test your maths skills. It will set you lots of knotty problems, and entertaining puzzles too.

Your computer is a very complicated machine. It can produce some quite amazing results. There are all sorts of computer games and puzzles available that tackle maths problems it would take you years to do on paper. To write these programs takes a great deal of skill and knowledge.

As a beginner you can easily get confused by all the instructions and information which you find in books and magazines. This book is here to help you make a start. It will explain as simply as possible how your computer can be used with numbers.

Although you could use this book if you've never used a computer, you will probably

find it uses words and terms that you don't understand. If so, you might like first of all to read the "Beginners' Micro Guide" to the Electron or BBC Micro, so that you have some idea how your computer works. It will make many ideas and words in this book easier to follow.

The programs in this book have been arranged so that you will learn, step by step, the words and commands of BASIC that you use to write programs for puzzles and calculations.

Don't worry if you don't understand exactly what each word or command means straight away. All you need to know for now is what it does!

Computer programs can be stored on cassette tape. With the MICRO MATES series, you can build up a collection of your own programs on tape.

The first few programs are very simple introductions to some ideas that you will explore later. You probably won't want to 'save' these, but you may well want to make a copy of some of the longer and more useful programs.

At the back of this book you will find a short section which tells you how to record and use your own programs.

The BBC Micro and the Electron are very similar, so for most of the programs in this book it doesn't matter which you have. But beware! The Electron does not have a MODE 7. Where a MODE 7 program will not work on the Electron, this is made clear; otherwise, if you are using an Electron, just type in MODE 6 instead.

Programming for Beginners

You will need to learn a number of BASIC words used for programming. It helps to keep a notepad and pencil handy for making notes. Before you begin, read through this chapter carefully. It will help you to avoid some common problems.

LINE NUMBERS

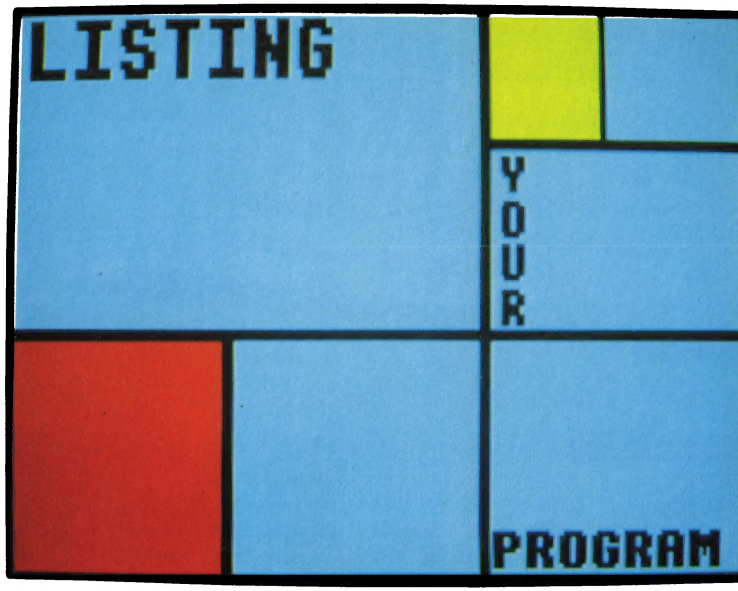
A computer program is essentially a list of numbered instructions. Each set of instructions begins with a line number. These usually start at ten and go up in tens. This is so that extra lines can be added later on. What matters is that the instructions in your program are arranged in the correct order.

THE RETURN KEY

When you type in a program line that is too long for one line on the screen you do not have to worry, it is automatically carried on to the next line. But when you have completed that set of instructions and are ready to type in the next line number, you MUST press the RETURN key.

THE DELETE KEY

If you make a mistake when typing something in, you can remove it by pressing the DELETE key at the bottom right hand corner of the keyboard.



RUNNING AND LISTING PROGRAMS

Once you have typed in a program, you have to tell your computer to RUN the program. To do this, type in RUN, and then press RETURN.

If all goes well, your program will work first time. If it doesn't, LIST your program and take another look at it.

If the program is a long one, start by pressing the CTRL and N keys at the same time. This stops the program running off the top of the screen when it lists. Then type in LIST and press RETURN. Now you can look at your program. To see the next page of a long program, just press the SHIFT key - the next page of the program will move up the screen.

SPACING AND PUNCTUATION BUGS

Your computer works by obeying instructions written in BASIC, and BASIC has some very strict rules. If you don't follow these, your programs will "crash" - that is, they won't work properly. When you break the rules of BASIC like this, it is called a SYNTAX ERROR.

Syntax errors can be caused by all sorts of small mistakes. When they happen, your computer will usually tell you at what line in the program it has not understood your instructions.

When you type in a computer program, you must make sure that you copy it in correctly. Even little things, such as not leaving a space between words or leaving too much space between certain symbols, can cause your programs to crash. Watch out for those punctuation marks, they are more important than they seem. The semi-colon symbol (;) has quite a different meaning to the colon symbol (:), and they should not be confused with each other.

HUNT THE BUG!

Your computer may have told you on which line the mistake has been made. LIST the program, look at the line and compare it with the same line in the printed listing. Sometimes the mistake (or 'bug' as they are sometimes called) is obvious. When you have found it, just type in the correct version of the line. Press RETURN and list the program again, and the bug will be gone.

Some bugs are not easy to find!

ZEROS AND O'S

A common mistake made by beginners is to confuse the 0 (zero) with the O (the letter O). Make sure particularly that you use the zero when typing in line numbers. Your computer will not accept an "O" for a "0".

CAPITAL LETTERS

Another easy mistake is to use small letters when CAPITALS must be used. All the BASIC keywords like PRINT, LIST, RUN etc must always be typed in capitals. If you're bothered by bugs, then check those capitals!

NEW

To get rid of an old program before typing in another one, just type in NEW and press RETURN.

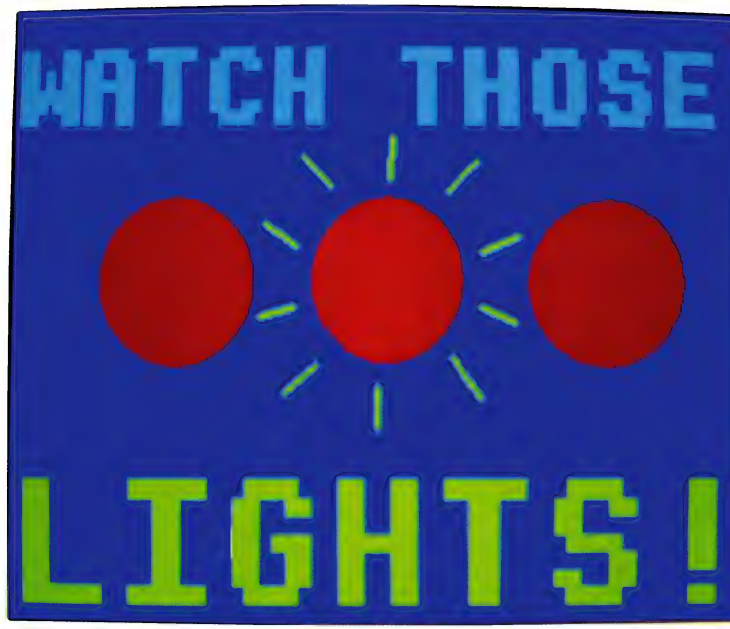
ABBREVIATIONS

You can shorten some instructions when you type in your programs, eg:-

PRINT can be shortened to P.
INPUT can be shortened to I.
LIST can be shortened to L.

When you list your programs these will be printed out in full.

A full list of the abbreviations that you can use is to be found in your User Guide.



WATCH THOSE LIGHTS!

At the bottom left hand corner of the keyboard are a group of three lights labelled "cassette motor", "caps lock" and "shift lock".

When the key marked CAPS LOCK is pressed, the caps lock light will either light up or switch off. With the caps lock light on, all letters typed in will be capitals. To obtain small, lower case letters, the caps lock light must be turned off by pressing the CAPS LOCK key again.

The same applies to the "shift lock" light and SHIFT LOCK key. With the shift lock light on, as well as all letters being capitals, the upper characters on the number and symbol keys are obtained.

Normally neither light need be on. Capital letters and the upper characters on the keys can be obtained by pressing the SHIFT key and the key chosen.

If you find yourself typing in some strange letters or symbols, check your lights!

THE BREAK KEY

Avoid pressing the BREAK key if you possibly can. If you do press it, you will see that the computer behaves as if it had just been switched on, and the standard message appears on the screen.To get back to your program, type in:-

```
OLD
```

and press RETURN. Then type in:-

```
LIST
```

Your program listing is displayed, and you have not lost your program.

CLEARING THE SCREEN

If you want to clear the screen while you are writing a program, type in:-

```
CLS
```

and press RETURN. This clears the screen and you can now list or run your program as normal. Pressing CTRL and L at the same time also clears the screen.

EDITING PROGRAMS

If you have made a mistake or you want to change part of a line, you can EDIT that line. This saves rewriting the whole line.

To do this, first move the screen cursor, the flashing line following the >, using the arrow keys at the top right of the keyboard. Move it until it reaches the beginning of the line you want to change.

Now press the COPY key. You will see that line being copied on to the screen. Keep your finger on the COPY key until you reach the part of the line that you want to change. You may now make whatever changes you like in the usual way. If you would like the rest of the line to stay as it is, then press the COPY key again until the rest of the line is copied out. Now press RETURN. The new version of the line goes into your program.

DELETING LINES

If you want to get rid of one line altogether, just type in the number of that line and then press the RETURN key. This removes that line from the memory.

To delete a group of lines, use the command DELETE. Type in the word DELETE followed by the first line number of the group of lines you wish to remove. Then type in a comma followed by the last line number of the group of lines to be removed. Finally press RETURN. A typical DELETE command might look like this:-

```
DELETE 100,500
```

MICRO MATHS

Your computer is a very powerful tool. As you will discover it can work out the answer to sums and calculations far faster than you!

Microcomputers have what is known as a memory. All the numbers that you use in a program or calculation are stored in this memory. But this memory is limited. This means that when you use very large numbers or decimal fractions your computer may not always treat them in the way you would expect.

For example, if you ask your computer to print the number .8888888888888 then what it will print is the number .888888889. If you were to ask it to print the number 10000000000 then what it would print would be this:-

```
1E10
```

which means ten raised to the power of ten.

These strange things happen because your computer just doesn't have enough space in its memory to store such big numbers. This probably won't affect the programs you use in this book, but be warned: computers are not always 100% accurate.

PROGRAM 1: MULTIPLICATION

Program 1 is a simple calculation program. It asks you to INPUT, or put in two numbers. It then multiplies these.

Lines 10 to 30 print the program's title. The use of the word PRINT on its own at Line 30 prints a blank line underneath.

Lines 40 and 60 print the questions asked by the program. Always try to make your questions clear and polite !

Lines 50 and 70 use the BASIC word INPUT. This allows you to store information. In this program, your two numbers are stored as the variables A and B.

Finally Line 80 prints the answer. Notice that the multiplication sign in BASIC is an asterisk (*).

Type in:-

```
 5 REM USING *
10 PRINT "MULTIPLICATION"
20 PRINT "--------------"
30 PRINT
40 PRINT"WHAT IS YOUR FIRST NUMBER ?"
50 INPUT A
60 PRINT"WHAT IS YOUR SECOND NUMBER ?"
70 INPUT B
80 PRINT A;"*";B;"=";A*B
```

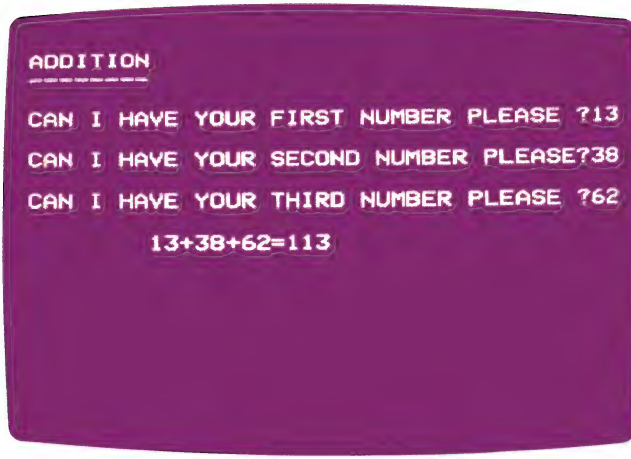
PROGRAM 2: ADDITION

Program 2 adds up any three numbers which you INPUT at Lines 70, 90 and 110. You can have as many variables as you like in a program, provided that you give each one a different name.

The instruction CLS at Line 10 clears the screen before the program runs.

Type in:-

```
 5 REM * ADDITION *
10 CLS
20 REM * 3 NUMBERS *
30 PRINT "ADDITION"
40 PRINT"________"
50 PRINT
60 PRINT"CAN I HAVE YOUR FIRST NUMBER
PLEASE ?"
70 INPUT A
80 PRINT"CAN I HAVE YOUR SECOND NUMBER
PLEASE ?"
90 INPUT B
100 PRINT"CAN I HAVE YOUR THIRD NUMBER
PLEASE ?"
110 INPUT C
120 PRINT A;"+";B;"+";C;"=";A+B+C
```


ADDITION
CAN I HAVE YOUR FIRST NUMBER PLEASE ?13
CAN I HAVE YOUR SECOND NUMBER PLEASE?38
CAN I HAVE YOUR THIRD NUMBER PLEASE ?62
13+38+62=113

THE REM STATEMENT

At Line 5 in these two programs you will see what is known as a REM statement. REM stands for REMARK. REM statements are put into programs to tell you about the program. This can be very helpful to anyone looking at the program.

PROGRAM 3: DIVISION

Program 3 shows the use of the division sign and of longer variable names.

Lines 70 and 90 use the variables number1 and number2. Although single letter variables could be used here, the longer names are clearer and make the listing easier to understand. Longer variable

names should be in lower case i.e not capital letters. Spaces are not permitted, so that the variable name 'number 1' would not be allowed.

Your computer does not use the traditional division sign. Instead it uses the / (or backslash) sign.

Type in:-

```
   10 REM * DIVISION USING / *
   20 CLS
   30 PRINT"DIVISION"
   40 PRINT"--------"
   50 PRINT
   60 PRINT "CAN I HAVE YOUR FIRST NUMBER
PLEASE ?"
   70 INPUT number1
   80 PRINT "CAN I HAVE YOUR SECOND NUMBER
PLEASE ?"
   90 INPUT number2
  100 PRINT number1;"divided by "; number2
;"="; number1/number2
```

PROGRAM 4: MOD & DIV

When you run Program 3 you will find that it treats all fractions as decimals. This can be awkward at times. Program 4 shows the use of the BASIC words MOD and DIV.

Lines 80 and 90 are the important lines here. Line 80 divides num1 by num2 and gives the whole number part of the calculation. At Line 90, MOD gives you the remainder.

RUN the program a few times.

Type in:-

```
   10 REM * MOD AND DIV *
   20 CLS
   30 PRINT"DIVISION WITH REMAINDERS"
   40 PRINT"-------------------------"
   50 PRINT
   60 INPUT"YOUR FIRST NUMBER PLEASE ",num1
   70 INPUT"YOUR SECOND NUMBER PLEASE ",num2
   80 PRINT; num1;" divided by ";
num2;"=";num1 DIV num2;
   90 PRINT" with ";num1 MOD num2;" remainder"
```

DIRECT COMMANDS

You can use your computer to work out some calculations directly, without using line numbers. Try typing in:-

```
PRINT 12*35
```

and press RETURN. You will be given the answer straight away.

PROGRAM 5: A COUNT

Program 5 makes use of a 'count'. This enables you to keep a running total.

Line 60 sets a variable,'total', to an initial value of 0. Any number INPUT while the program runs is added to 'total'.

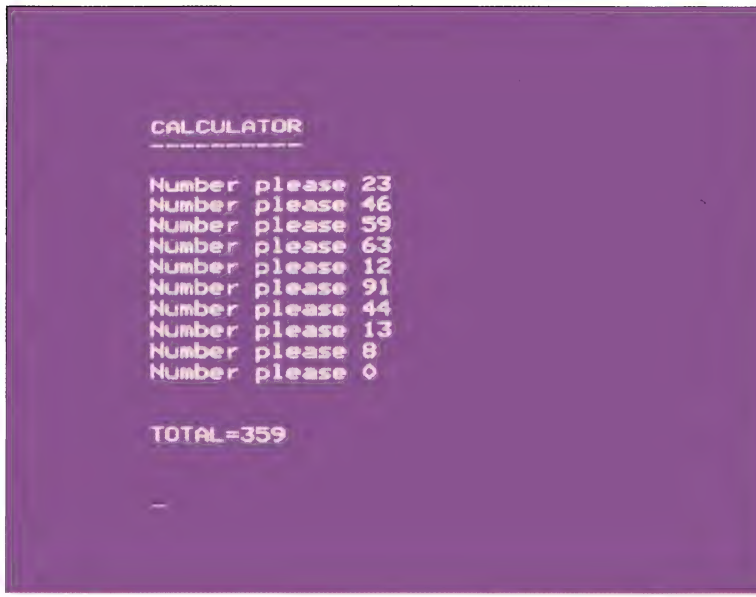
Line 80 INPUTs the number you type in and Line 90 adds this number to 'total'. Line 100 checks to see if 0 has been INPUT. If it has then the program moves to Line 110 and prints the final value of 'total'. If 0 is not INPUT then the program returns to Line 70.This enables you to add a further number.

You can add up as many numbers as you like using this program. When you are ready to find out the total, just INPUT 0.

Type in:-

```
10 REM *LOOP & COUNT*
20 CLS
```

```
 30 PRINT"CALCULATOR"
 40 PRINT"__________"
 50 PRINT
 60 LET total=0
 70 PRINT "Number please"
 80 INPUT number
 90 total=total+number
100 IF number=0 THEN GOTO 110 ELSE 70
110 PRINT "TOTAL =";total
```



PROGRAM 6: REPEAT..UNTIL LOOP

This program does the same calculation as Program 5 but uses a different method.

Like Program 5, it allows certain sections of your program to be repeated. This is known as a LOOP. Because it uses the words REPEAT and UNTIL, the loop used here is known as a REPEAT..UNTIL loop.

Type in:-

```
 10 REM * REPEAT..UNTIL *
 20 CLS
 30 PRINT "CALCULATOR"
 40 PRINT "__________"
 50 PRINT
 60 score=0
 70 REPEAT
 80 INPUT "Number please ",num
 90 score=score+num
100 UNTIL num=0
110 PRINT score
```

LINE NUMBERING

To save time when typing programs in, you can use the command AUTO. This automatically prints line numbers. Before writing your next program, type in:-

```
AUTO
```

and press RETURN. The line number 10 appears automatically, then (after you have pressed RETURN) 20, and so on. When you have finished typing in your program, press ESCAPE. This stops the automatic line numbering.

PROGRAM 7: ADDITION QUIZ

Program 7 is a longer program that uses various features of BASIC. It sets six addition questions for you to answer.

Lines 20 to 70 are a friendly way to start a program !

Line 90 sets up a count to keep score. Correct answers gain a mark. Incorrect answers add a mark to the computer's score.

Lines 80, 210, 330 and 370 call up a PROCEDURE. This is a part of a program which can be used several times. The procedure is defined at Lines 390 to 410. It is used to slow down the running of the program.

Line 100 sets up a FOR..NEXT loop to ask six questions. If you want to be asked more than six, increase this number.

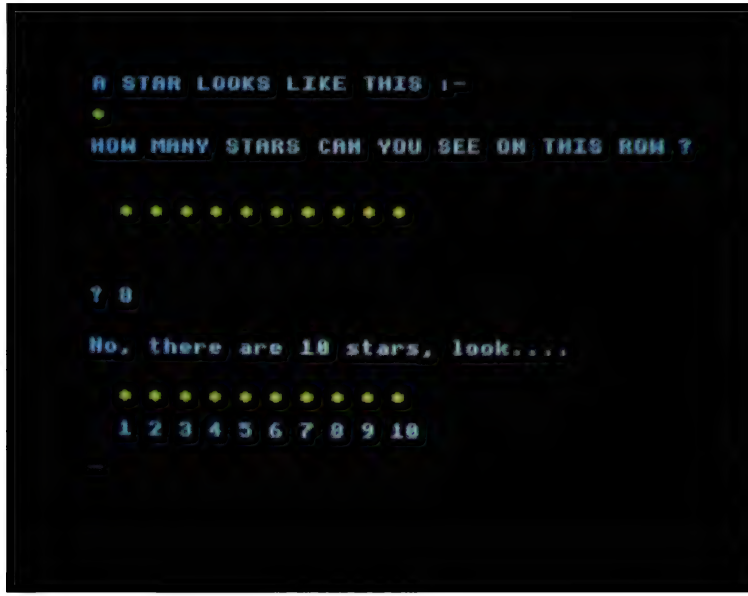
Line 140 uses the BASIC word RND to pick numbers at random for the quiz. If you want larger numbers, increase the numbers in brackets.

Lines 190 and 200 check your answer. If correct, the program goes to the SUBROUTINE at Line 310. If incorrect, the program goes to the subroutine at Line 350. A subroutine is also a part of a program that may be used several times over.

Finally Lines 240 to 300 print the final score and end the program.

Type in:-

```
 10 REM * Addition Quiz *
 20 MODE 7
 30 PRINT "What is your name ?"
 40 INPUT name$
 50 PROCpause
 60 CLS
 70 PRINT "Hello ";name$
 80 PROCpause
 90 score=0:beeb=0
100 FOR question=1 TO 6
110 CLS
120 PRINT "MATHS TEST"
130 PRINT "----------"
140 X=RND(10)+5:Y=RND(10)+5
150 PRINT '"WHAT IS THE ANSWER TO THIS
PROBLEM ?"
160 PRINT
170 PRINT X;"+";Y;" ?"
180 INPUT ans
190 IF ans= X+Y THEN GOSUB 310
200 IF NOT (ans =X+Y) THEN GOSUB 350
210 PROCpause
220 NEXT question
230 CLS
240 PRINT "FINAL SCORE "
250 PRINT "___________"
260 PRINT
270 PRINT name$;" ";score
280 PRINT "Computer ";beeb
290 PRINT
300 END
310 PRINT '" Correct. One mark scored."
320 score=score+1
330 PROCpause
340 RETURN
350 PRINT '" Incorrect answer."
360 beeb=beeb+1
370 PROCpause
380 RETURN
390 DEF PROCpause
400 FOR Z=1 TO 1500:NEXT Z
410 ENDPROC
```



PROGRAM 8: COUNTING STARS

Program 8 prints a row of stars on the screen and asks you to count them.

This program uses MODE 1. There are eight display modes on the BBC Micro and seven on the Electron, each offering a different range of colours and special features.

Line 40 sets up a loop to give you five turns. For more turns simply increase this number. Line 60 uses the COLOUR instruction to print a yellow 'star'. COLOUR 2 stands for yellow in MODE 1. The text is displayed in white; the code for white is COLOUR 3.

Line 90 selects at random the number of stars to be displayed. To increase the range of numbers picked, you increase the number in brackets.

The subroutine at Line 320 then prints out the row of stars using the instruction TAB. This spaces out the display along the line.

Lines 150 checks your answer. If correct, the program moves to the subroutine at Line 200. If incorrect, the program moves to the subroutine at Line 230. This prints out the same number of stars, and numbers each one using the TAB instruction.

Finally Lines 180 and 190 print out the score and end the program.

Type in:-

```
   10 REM *STARS*
   20 MODE 1
   30 score=0
   40 FOR go=1 TO 5
   50 PRINT '"A STAR LOOKS LIKE THIS :-"
   60 PRINT:COLOUR 2:PRINT "*":COLOUR 3
   70 PRINT' "HOW MANY STARS CAN YOU SEE
 ON THIS ROW ?"
   80 COLOUR 2
   90 LET stars=RND(7)+3
  100 PRINT:PRINT
  110 GOSUB 320
  120 COLOUR 3:PRINT:PRINT
  130 PRINT:PRINT:PRINT
  140 INPUT answer
  150 IF answer=stars THEN GOSUB 200
 ELSE GOSUB 230
  160  PRINT:PRINT:FOR Z = 1 TO 800:
 NEXT Z: CLS
  170 NEXT go
  180 CLS: PRINT' "You scored ";score;"
 out of 5 "
  190 PRINT:END
  200 PRINT'' "Correct, there are ";
stars;" stars"
  210 score=score+1
  220 RETURN
```

```
  230 PRINT''"No, there are ";stars;"
stars, look...."
  240 PRINT:PRINT
  250 GOSUB 320
  260 PRINT:PRINT:
  270 FOR N=1 TO stars
  280 PRINT TAB(N*2);N;
  290 FOR Z=1 TO 400:NEXT Z
  300 NEXT N
  310 RETURN
  320 FOR N=1 TO stars
  330 FOR Z=1 TO 400:NEXT Z
  340 PRINT TAB(N*2);"*";
  350 NEXT N
  360 RETURN
```

PROGRAM 9: HOW MANY HOUSES ?

Program 9 is a more adventurous version of the last program. It uses MODE 2 graphics. It prints a number of houses on the screen, and you have to count them correctly.

This program works with 'user defined graphics'. Lines 30 and 40 define a house shape, stored as CHR$(240) and CHR$(241). This shape is then put in the procedure 'house', defined at Lines 300 to 320.

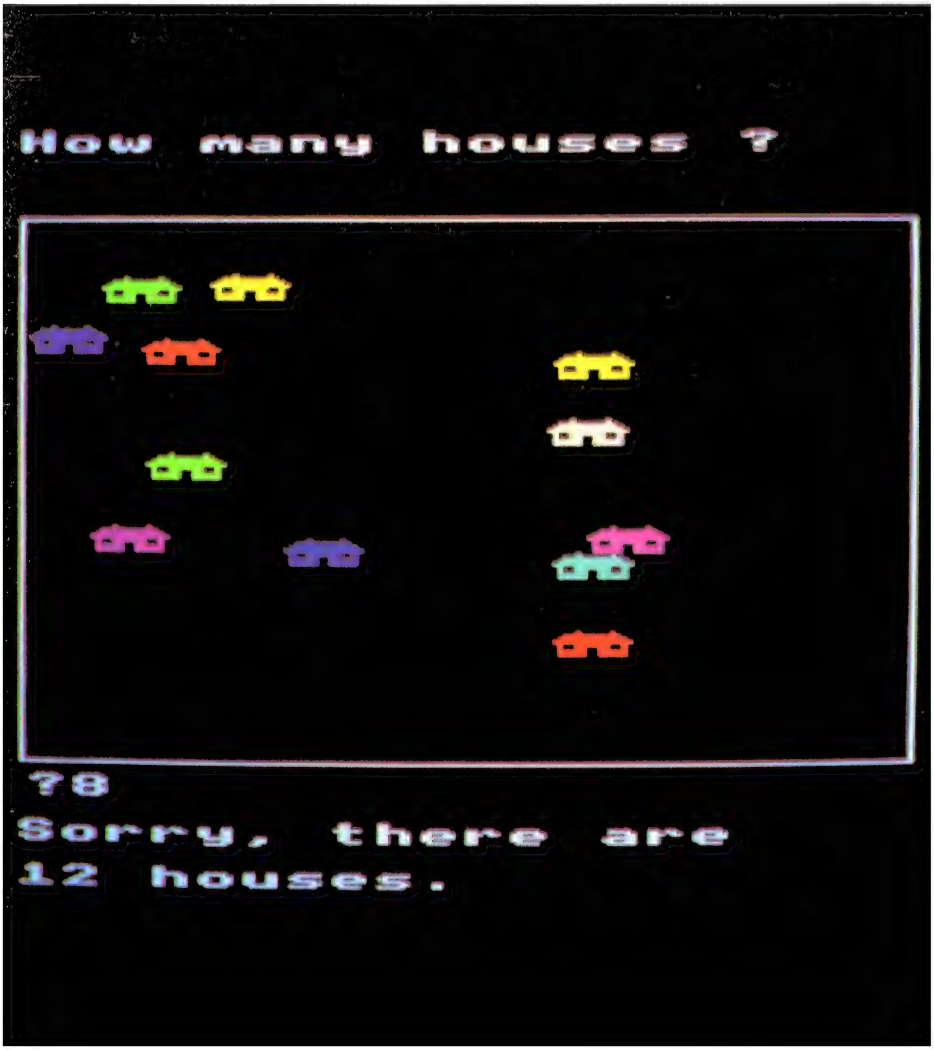
A loop is set up at Line 60 to set the number of turns in the quiz. A sound effect is added at Lines 100 to 120.

Line 130 calls up the procedure 'boundary', defined at Lines 330 to 370. This draws a frame around the area where the houses will be printed.

Lines 140 to 190 use a loop to print out the houses at various places on the screen.

Line 170 picks at random the X and Y co-ordinates for the position of each house. This point is then established using the instruction MOVE at Line 180. The instruction VDU 5 which follows, enables shapes and letters to be plotted on the graphics display grid.

Finally, at Lines 200 to 290, the answers are INPUT and checked, your score is added up and the final result printed out.



Type in:-

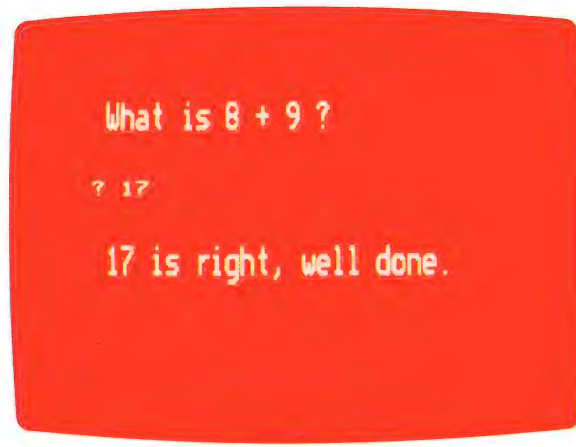
```
   10 REM * VILLAGE *
   20 MODE 2
   30 VDU 23,240,4,31,63,127,39,38,62,62
   40 VDU
23,241,16,248,252,254,228,100,124,124
   50 score=0:question=5
   60 FOR turn=1 TO 5
   70 CLS
   80 PRINT "How many houses ?"
   90 village=RND(14)
  100 FOR n=150 TO 0 STEP-5
  110 SOUND 1,-10,n,1
  120 NEXT n
  130 PROCboundary
  140 FOR houses =1 TO village
  150 col=houses:IF col>7 THEN
col=houses-7
  160 GCOL0,col
  170 X=RND(1000)+100:Y=RND(500)+350
  180 MOVE X,Y:VDU5:PROChouse
  190 NEXT houses
  200  MOVE 20,180:VDU 5:GCOL0,7: INPUT
answer
  210 IF answer=village THEN PRINT'
"Correct": score=score+1: GOTO 240
  220 IF NOT(answer=village) THEN PRINT'
"Sorry , there are "
  230 PRINT';village;" houses."
  240 FOR Z=1 TO 4000:NEXT
  250 NEXT turn
  260 CLS: PRINT "QUIZ OVER"
  270 PRINT'"You scored ";score
  280 PRINT'"out of ";question
  290 PRINT :END
  300 DEF PROChouse
  310 PRINT;CHR$(240);CHR$(241)
  320 ENDPROC
  330 DEF PROCboundary
  340 MOVE 10,200:DRAW 1279,200
  350 DRAW 1279,900: DRAW 10,900
  360 DRAW 10,200
  370 ENDPROC
```

PROGRAM 10: DOUBLE HEIGHT LETTERS

Program 10 is another addition quiz. If your computer is a BBC Micro, it will give you an easy-to-read screen display in double height letters. If you have an Electron, you can still do the quiz. Type it in in MODE 6, and leave out the instruction CHR$ and Lines 90, 170, 230 and 300.

For BBC Micro users, the most important lines are Lines 80 and 90, 160 and 170, 220 and 230, 290 and 300. Here the instruction PRINT CHR$(141) prints the text that follows in double height letters. This instruction has to be repeated on the following line of the program, otherwise only the top half of your text gets printed.

MODE 7 is known as the TELETEXT mode because it is the mode used to print Teletext television displays. MODE 7 differs from the other modes and requires different commands and instructions. Section 28 of the BBC Micro's USER GUIDE deals with Teletext mode in more detail.



Type in:-

```
   10 REM * Double Height Display *
   20 MODE 7
   30 score=0
   40 FOR turn=1 TO 5
   50 num1=RND(5)+5
   60 num2=RND(5)+5
   70 PRINT :PRINT
   80 PRINT CHR$(141);"What is ";num1;" +
";num2;" ?"
   90 PRINT CHR$(141);"What is ";num1;" +
";num2;" ?"
  100 PRINT :PRINT
  110 INPUT answer
  120 IF answer=num1+num2 THEN PROCcorrect
ELSE PROCwrong
  130 NEXT turn
  140 CLS
  150 PRINT :PRINT
  160 PRINT CHR$(141);"SCORE = ";score;"
OUT OF 5"
  170 PRINT CHR$(141);"SCORE = ";score;"
OUT OF 5"
  180 END
  190 DEF PROCcorrect
  200 score=score+1
  210 PRINT :PRINT
  220 PRINT CHR$(141); answer ; " is
right, well done."
  230 PRINT CHR$(141); answer ; " is
right, well done."
  240 FOR N=1 TO 3000: NEXT
  250 CLS
  260 ENDPROC
  270 DEF PROCwrong
  280 PRINT :PRINT
  290  PRINT CHR$(141);"No, you are wrong.
The answer is ";num1+num2
  300   PRINT CHR$(141);"No, you are
wrong. The answer is ";num1+num2
  310 FOR N=1 TO 5000: NEXT
  320 CLS
  330 ENDPROC
```

PROGRAM 11: A DIFFERENT ROUTINE

Instead of the procedure 'wrong' that was used in Program 10, this program has a procedure that gives you more help. It tells you whether the wrong answer was too big or too small. Add it to Program 10. Electron users should leave out Lines 320 and 360.

To remove the procedure 'wrong' in Program 10, type in the command:-

```
DELETE 270,330
```

and press RETURN. You can now enter the new procedure described in Program 11.

This new procedure checks your answer at Lines 290 and 300.If your answer is larger than the correct answer then the program moves to Line 310. If smaller, the program moves to Line 350.

If a wrong answer has been given, you can have a second go. At Lines 330 and 370 the program returns to Line 70 (of Program 10) and the question is set again.

Type in:-

```
270 DEF PROCwrong
280 PRINT :PRINT
290 IF answer>num1+num2 THEN 310
300 IF answer<num1+num2 THEN 350
310 PRINT CHR$(141);"No, ";answer;" is
too big, try again."
320 PRINT CHR$(141);"No, ";answer;" is
too big, try again."
325 FOR N=1 TO 3000: NEXT
330 CLS:GOTO 70
350 PRINT CHR$(141);"No, ";answer;" is
too small, try again."
```

```
  360 PRINT CHR$(141);"No, ";answer;" is
too small, try again."
  365 FOR N=1 TO 3000: NEXT
  370 CLS:GOTO 70
  380 FOR N=1 TO 5000: NEXT
  390 ENDPROC
```

PROGRAM 12: YES OR NO ?

Program 12 is another procedure. You can add it to the end of most programs. The line numbering begins at 400 so that it can be added to the last program. Electron users should leave out Lines 440 and 480.

When you have typed this into your program you will have to add an extra line. Just before the final END statement, put in a line like this:-

    line no. PROCagain

If you want another go at running the program, just type in Y (in capitals) when the program asks you if you wish to have another go.

Type in:-

```
  400 PROCagain
  410 END
  420 DEF PROCagain
  430 PRINT CHR$(141);"Do you want another
go, Y or N ?"
  440 PRINT CHR$(141);"Do you want another
go, Y or N ?"
  450 INPUT ans$
  460 IF NOT (ans$="Y")THEN 470 ELSE 10
  470 PRINT CHR$(141); "Thankyou for
playing the quiz."
  480 PRINT CHR$(141); "Thankyou for
playing the quiz."
  490 ENDPROC
```

RENUMBERING YOUR LINES

If you add lines of your own to a program, the line numbering system can get untidy. Try typing in the instruction:-

```
RENUMBER 10,10
```

This renumbers the lines in your program, starting at 10 and leaving a gap of 10 between each line number.

PROGRAM 13: PERCENTAGE CALCULATOR

Program 13 works out the percentage of any given number. It uses MODE 4. Line 30 alters the colour of your display.

Lines 60 to 120 ask the questions necessary for the calculation.

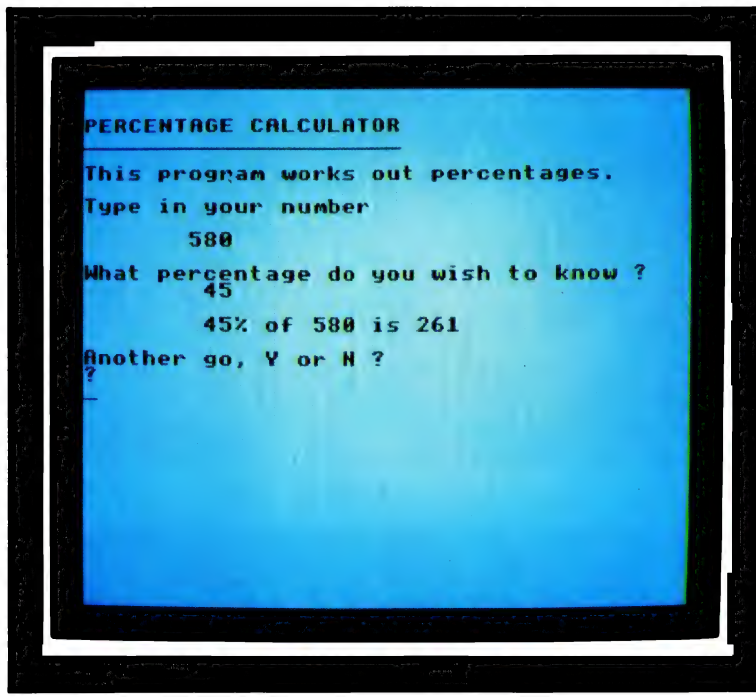
Line 130 works out the required percentage, INPUT at Line 120, of your number, INPUT at Line 90.

Line 140 prints out the answer and Lines 150 to 170 allow you, if you wish, to have another go.

Type in:-

```
10 REM * PERCENTAGES *
20 MODE 4
30 COLOUR 129:COLOUR 0: CLS
```

```
   40 PRINT''"PERCENTAGE CALCULATOR "
   50 PRINT"_____ "
   60 PRINT'"This program works out
percentages."
   70 PRINT'"Type in your number "
   80 PRINT
   90 INPUT number
  100 PRINT
  110 PRINT "What percentage do you wish
to know ?"
  120 INPUT percentage
  130 LET ans=number*(percentage/100)
  140 PRINT percentage;"%";" of ";number;"
is ";ans
  150 PRINT'"Another go, Y or N ?"
  160 INPUT go$
  170 IF go$="Y" THEN 20 ELSE PRINT'"BYE
BYE"
```



PROGRAM 14: CURRENCY CONVERTER

Program 14 works out how many dollars you would get for any number of pounds sterling.

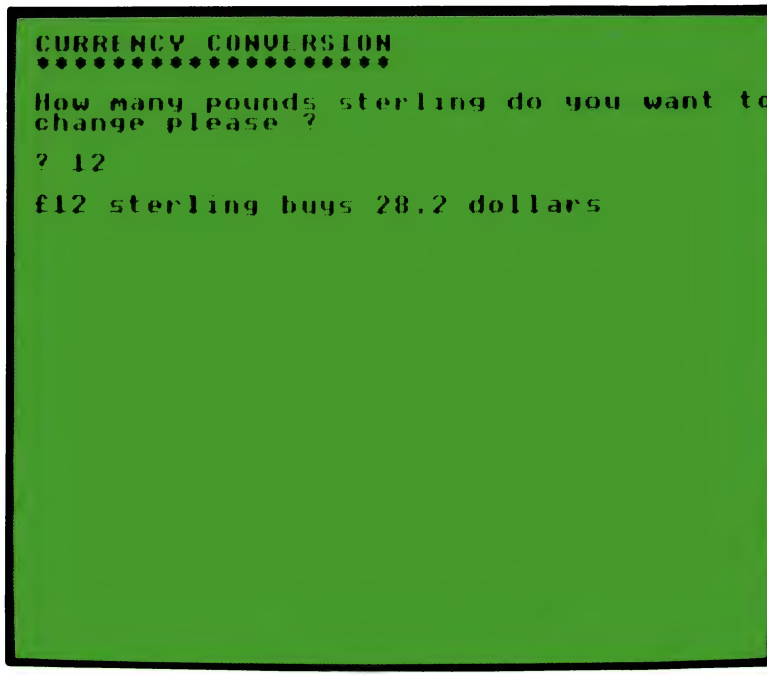
Lines 30 and 40 change the colours used in the display. Lines 60 and 70 print a title for the program.

Lines 80 and 90 ask you how many pounds sterling you wish to change. The current exchange rate is given at Line 100. This can be altered to suit the rate. Line 150 prints out the results, while Lines 110 to 130 add a sound effect.

This is the kind of program that could be modified to suit any number of currencies.

Type in:-

```
   10 REM * CONVERSION PROG *
   20 MODE 4
   30 COLOUR 128
   40 VDU 19,128,2,0,0,0
   50 COLOUR 1
   60 VDU 19,1,0,0,0,0
   70 CLS
   80 PRINT' " CURRENCY CONVERSION"
   90 PRINT " ********************"
  100 PRINT'" How many pounds sterling  do
you want to change please ?"
  110 PRINT'" ";:INPUT pounds
  120 LET dollar=pounds*1.375
  130 FOR n=100 TO 250 STEP10
  140 SOUND1,-15,n,1
  150 NEXTn
  160 FORz=1 TO 500: NEXT
  170 PRINT' " £";pounds;" sterling buys
";dollar;" dollars"
```



PROGRAM 15: METRIC CONVERSION

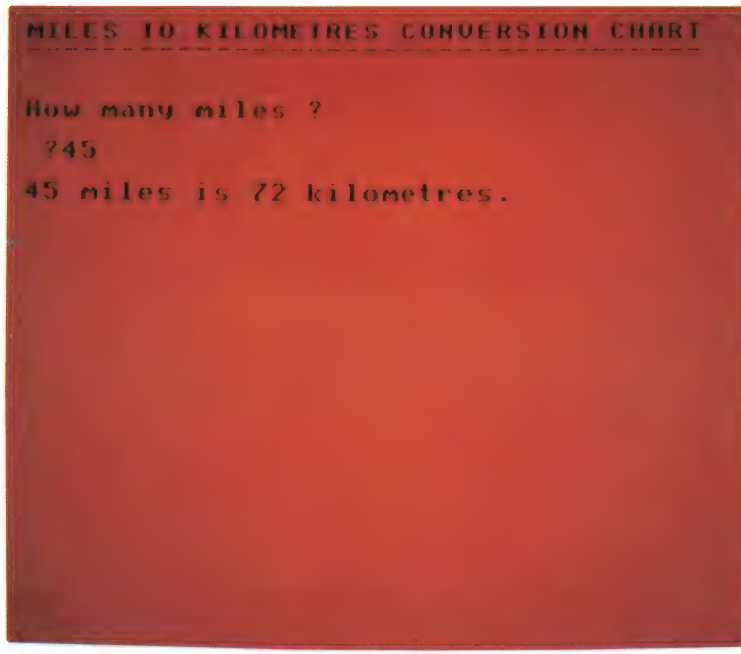
Program 15 converts miles to kilometres. Lines 20 to 70 set up the screen display. Lines 80 and 90 ask for the relevant information.

Line 100 prints the answer, having multiplied the number of miles by 1.6 to calculate the number of kilometres.

Line 110 is used here to print a blank line before the cursor appears.

Type in:-

```
   10 REM * Miles to KM *
   20 MODE 4
   30 COLOUR 128
   40 VDU 19,128,1,0,0,0
   50 COLOUR 1
   60 VDU 19,1,0,0,0,0
   65 CLS
   70 PRINT ' " MILES TO KILOMETRES
CONVERSION CHART"
   80 PRINT
   90 PRINT''" How many miles ?"
  100 PRINT'" ";: INPUT miles
  110 PRINT
  120 PRINT" ";miles;" miles is "; miles
*1.6;" kilometres."
  130 PRINT
```



PROGRAM 16: SHOPPING LIST

Program 16 allows you to enter a list of prices and adds these up.

Line 80 asks you to INPUT the cost of each item. Line 100 asks you to INPUT the number of items that you are buying.

Line 120 works out and prints the sum cost of those items. Line 130 adds this to the 'total' set at Line 30.

Line 160 checks whether you have anything more to add to the list.

Lines 170 to 200 check the cost and divide this up, if necessary, into pounds and pence. The BASIC word INT used at Line 180

rounds down fractional numbers.It is used here to work out the number of pounds spent.

Then Lines 210 to 250 are used to print the final total.

Type in:-

```
   10 REM * SHOPPING LIST *
   20 MODE 7
   30 total=0
   40 PRINT'"SHOPPING LIST"
   50 PRINT "-------------"
   60 PRINT' "This program adds up a list
of prices."
   70 PRINT' "Cost of item in pence ?"
   80 INPUT cost
   90 PRINT "Number of items at ";cost;"
pence ?"
  100 INPUT items
  110 sum=cost*items
  120 PRINT  "Cost of ";items;" items at
";cost;" pence = ";sum;" pence"
  130 total=total+sum
  140 PRINT'"Any more, Y or N ?"
  150 INPUT more$
  160 IF more$="Y" THEN 70 ELSE 170
  170 IF total>100 THEN 180 ELSE 250
  180 pounds=INT(total/100)
  190 remainder=total-(pounds*100)
  200 IF remainder <10 THEN 230 ELSE 210
  210 PRINT "TOTAL COST = £ ";pounds;".";
remainder
  220 END
  230 PRINT "TOTAL COST =
£";pounds;".0";remainder
  240 END
  250 PRINT  "TOTAL COST = ";total;"
pence"
```

PROGRAM 17: PRIORITIES

If someone asks you what is two times two plus four, is your answer eight or twelve? It could be either, depending on whether you do the addition or the multiplication first.

Your computer has a simple rule for calculations involving several sums:-

Multiplication and division come first.
Addition and subtraction come next.

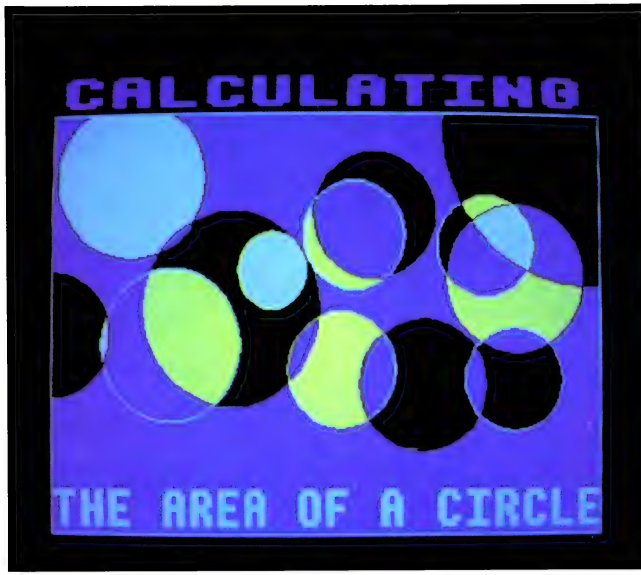
If you want your computer to work out addition and subtraction first, you need to enclose these sums in brackets.

Program 17 shows how this works. Line 20 produces the answer 8. It works out 2 times 3 first and then adds 2. However Line 30 produces the answer 12 because the sum in brackets, 2 plus 2 is worked out first. This is then multiplied by 3 to give the answer 12.

Lines 50 and 60 show a similar example.

Type in:-

```
10 REM * Priorities *
20 PRINT 2+2*3
30 PRINT(2+2)*3
50 PRINT 8-2*2
60 PRINT (8-2)*2
```



PROGRAM 18: THE AREA OF A CIRCLE

BBC Basic contains many words connected with mathematical formulae. Program 18 demonstrates the use of "PI".

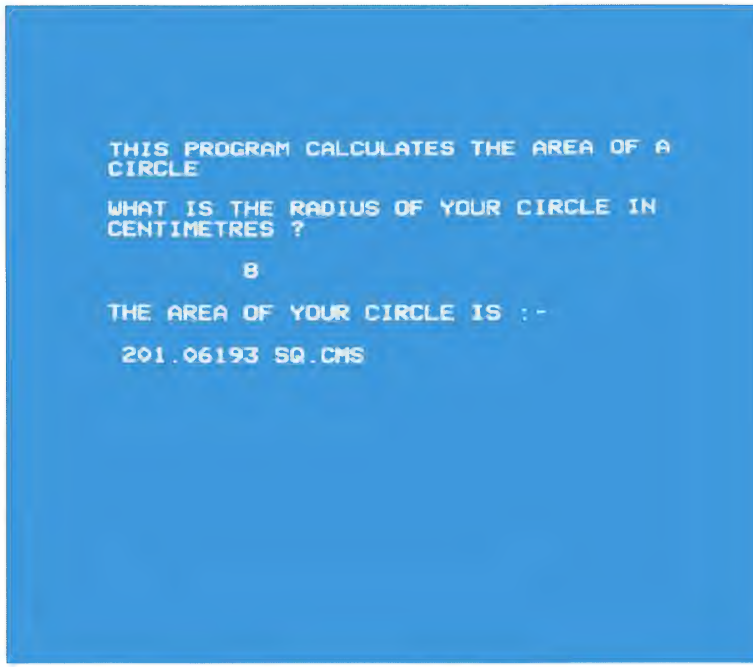
PI is used to calculate the area of a circle.

Line 50 asks you to input the radius of a circle, measured in centimetres.

Line 60 then uses the formula "PI*R squared" to calculate the area of the circle. The squared sign is the ^ symbol, just below the BREAK key.

Type in:-

```
   10 REM * CIRCLE AREA *
   20 MODE 7
   30 PRINT'"THIS PROGRAM CALCULATES THE
AREA OF A   CIRCLE"
   40 PRINT'"WHAT IS THE RADIUS OF YOUR
CIRCLE IN    CENTIMETRES ?"
   50 INPUT 'radius
   60 area=PI*radius^2
   70 PRINT' "THE AREA OF YOUR CIRCLE IS
:-"
   80 PRINT' area; " SQ.CMS"
```


THIS PROGRAM CALCULATES THE AREA OF A CIRCLE
WHAT IS THE RADIUS OF YOUR CIRCLE IN CENTIMETRES ?
8
THE AREA OF YOUR CIRCLE IS :-
201.06193 SQ.CMS

LISTING YOUR PROGRAMS

Reading program listings in some modes can be quite hard work. MODE 7 on the BBC Micro, and MODE 6 on the Electron are the easiest modes to read. You can get all your programs listed in these modes by using the red USER DEFINED KEYS at the top of your keyboard. Type in one of these lines:-

```
*KEY 0 MODE7|M LIST|M
*KEY 0 MODE6|M LIST|M
```

and press RETURN. Now press red key f0. You only have to press this key now whenever you want a MODE 6 or 7 listing.

PROGRAM 19: CORNER SHOP

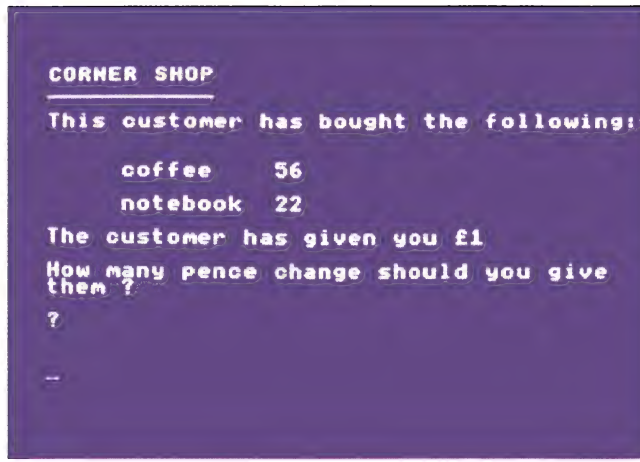
Program 19 puts you in charge of a corner shop. A customer comes in and makes several purchases. You have to add up the bill and work out how much change to give.

Line 50 is a DIM or 'dimension' statement. This tells you the number of items in the shop (in this program, 15). The number of prices is also stated.

Line 70 sets up a loop to 'read' the DATA statements at Lines 340 to 380. These contain the names and prices of the goods on sale. These can be changed, as can the number of items. The number of items on your list should agree with the number mentioned in your DIM statement.

Lines 90 to 160 pick a selection of goods at random and list these and their prices. To increase the number of items bought, change the number at Line 100.

Lines 170 to 200 work out the total and the correct change. Line 170 ensures that the number of pounds is always greater than the total bill. Line 190 asks you to give what you think is the correct sum. This is then compared with the correct answer at Line 220.


CORNER SHOP
This customer has bought the following:-
coffee 56
notebook 22
The customer has given you £1
How many pence change should you give them ?
?

Type in:-

```
   10 REM * Change Test *
   20 MODE 4
   30 PRINT' "CORNER SHOP"
   40 PRINT  "___________"
   50 DIM item$(15),cost(15)
   60 FOR N=1 TO 15
   70 READ item$(N),cost(N)
   80 NEXT N
   90 PRINT' "This customer has bought the
following  :-"
  100 number=RND(4)+1
  110 total=0
  120 FOR N=1 TO number
```

```
  130 rand=RND(15)
  140 PRINT' TAB(5); item$(rand);TAB(15);
cost(rand)
  150 total=total+cost(rand)
  160 NEXT N
  170 cash=INT(total/100)+1
  180 PRINT' "The customer has given you
£";cash
  190 PRINT'"How many pence change should
you   give them ?"
  200 correct=(cash*100)-total
  210 PRINT :INPUT change
  220 IF change=correct THEN PROCright
ELSE PROCwrong
  230 DEF PROCright
  240 PRINT'"Correct, well done."
  250 GOTO 310
  260 ENDPROC
  270 DEF PROCwrong
  280 PRINT"No, you have made a mistake.
The bill   came to ";total
  290 PRINT "Have another try": GOTO 210
  300 ENDPROC
  310 INPUT' "Another go, Y or N", ans$
  320 IF ans$="Y" THEN 20 ELSE PRINT
'"Thankyou, bye bye."
  330 END
  340 DATA papers,18,sweets,35,milk,21
  350 DATA comic,23,postcard,15,bread,28
  360 DATA pen,25,pencil,15,notebook,22
  370 DATA battery,34,magazine,45,ink,32
  380 DATA butter,35,tea,47,coffee,56
```

PROGRAM 20: CORNER SHOP RAFFLE

Now add these lines to Program 19:-

```
    100 number= RND(10)+1
    242 raffle=INT(total/50)
    244 PRINT "Customers get a Christmas
  Hamper Raffle Ticket for every 50 pence
  they spend. How many tickets should you
  give ?"
    246 INPUT tickets
    248 IF tickets=raffle THEN PRINT
  "Correct" ELSE PRINT "Wrong, ";raffle
```

PROGRAM 21: ADDITION SERIES

Program 21 asks you to look at a row of numbers and to say what you think the following three numbers in that series should be.

The main program is divided into two procedures. The first procedure, which begins at Line 50, introduces the program.

The main part of the program, the quiz, begins at Line 150.

Line 160 decides the range of the numbers used. This can be made smaller or larger to suit. Line 170 sets up a loop to print out the first four numbers of the series.

Lines 210 and 230 ask you to INPUT your answers. When the program is running you need to press the RETURN key after each answer has been entered.

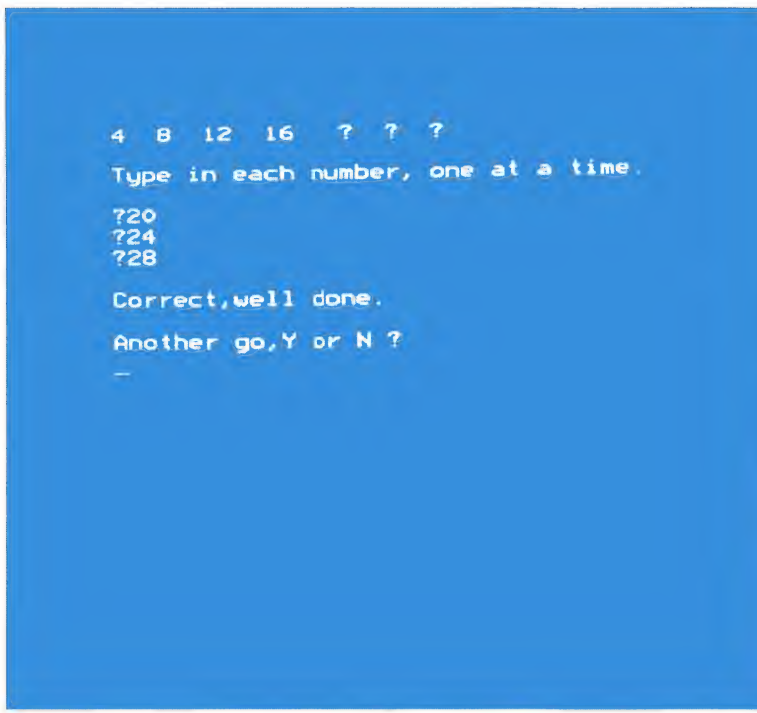
Line 240 checks to see if all three answers are correct. If so the program moves to Line 260, to the procedure 'correct'. If the answers are not correct then the program moves to the procedure 'wrong' at Line 300. This procedure allows you to have another go at the question and offers some help.

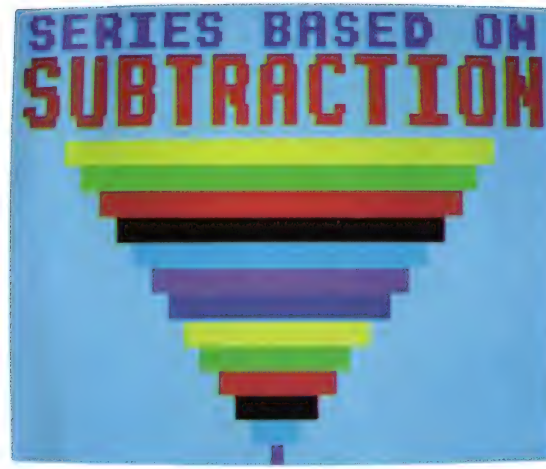
Finally Lines 350 to 380 offer the chance to play the game again.

Type in:-

```
   10 MODE 7
   20 PROCintroduction
   30 PROCquiz
   40 END
   50 DEF PROCintroduction
   60 PRINT' "SERIES BASED ON ADDITION"
   70 PRINT'"This program will give you 4
numbers."
   80 PRINT'"You have to work out the next
3,like so,"
   90 PRINT 'TAB(7);"2  5  8  11  ?  ?  ?"
  100 PRINT'"The answer would be 14,17 and 20"
  110 PRINT'"We have added 3 on to each number
to get the next number in the series"
  120 INPUT '"Are you ready to have a go,Y or N
",ans$
  130 IF ans$="Y"THEN ENDPROC ELSE
PRINT"BYE":END
  140 ENDPROC
  150 DEF PROCquiz
  160 CLS:number=RND(12):PRINT :PRINT
  170 FOR n=1 TO 4
  180 PRINT;number*n;"  ";
  190 NEXT n
  200 PRINT " ?  ?  ?"
  210 PRINT'"Type in each number, one at a
time."
  220 num1=number*4+number:num2=num1 +number:
num3=num2+number
  230  INPUT ans1,ans2,ans3
```

```
  240 IF ans1=num1 AND ans2=num2 AND ans3=num3
THEN PROCcorrect ELSE PROCwrong
  250 ENDPROC
  260 DEF PROCcorrect
  270 PRINT "Correct,well done."
  280 PROCagain
  290 ENDPROC
  300 DEF PROCwrong
  310 PRINT'"No your answer is not right, you
need   to add ";number;" to each number to find
the     next number in the series.",
  320 PRINT"Try again.":FOR Z=1 TO 4000:NEXT Z
  330 GOTO 230
  340 ENDPROC
  350 DEF PROCagain
  360 INPUT'"Another go,Y or N,"ans$
  370 IF ans$="Y" THEN PROCquiz ELSE PRINT'"
BYE":END
  380 ENDPROC
```





PROGRAM 22: A MINUS SERIES

Program 22 consists of a series of lines which can be inserted into Program 21 to test your subtraction skills.
Program 21 should be saved on tape first. Then load it again, and type in these new lines (they will automatically override the corresponding lines of Program 21). Save this program too, if you want to keep a copy of both. There are instructions on loading and saving at the back of the book.

Type in:-

```
   60 PRINT' "SERIES BASED ON
SUBTRACTION"
   90 PRINT 'TAB(7);"24 21 18 15  ?  ?
?"
  100 PRINT'"The answer would be 12,9
and 6."
  110 PRINT'"We have taken 3 from each
number to get the next number in the
series"
```

```
  160 CLS:number=RND(8):PRINT :PRINT
  170 FOR n=8 TO 4 STEP-1
  220 num1=number*4-number :num2=
num1-number :num3=num2-number
  310 PRINT'"No your answer is not  right,
you  need to take ";number;" from each
number to find the      next number in the
series."
```

PROGRAM 23: BIG NUMBERS

Program 23 demonstrates how quickly your computer can run out of memory space. It starts by taking X as being equal to 2. It then increases this by raising it to the power of 2, or squaring it. The resulting number then becomes the new value of X, and the program moves back to Line 20.

When the program is run you will soon see the error message 'Too big at Line 20'. Computers which can handle big numbers are known as NUMBER CRUNCHERS!

Type in:-

```
10 LET X=2
20 LET Z=X^2
30 PRINT Z
45 X=Z
50 GOTO 20
```

PROGRAM 24: TABLES TESTER

Program 24 tests your multiplication table skills. It asks you to select a number. It prints out the table for this number, then tests your knowledge of that table. If you get some of the answers wrong then it allows you to have another try.

Lines 70 to 100 use a loop to print out the table. Line 90 slows the display down so that it runs more slowly.

When you have studied the table for long enough, press key Y and the test begins. Lines 180 to 230 then print out each line of the table and ask you to type in your answer. If it is correct, the program

moves to the procedure 'right'and your score is increased by one. If incorrect, the program moves to the procedure 'wrong'. This offers a bit of helpful advice and gives you another chance.

If you get all the answers right the computer plays you a tune, produced at Lines 250 to 270.

Finally the procedure 'again' asks you if you want another turn.

You can choose any number you like, even decimals, within the usual limits.

Type in:-

```
   10 REM *TABLES TESTER *
   20 MODE 7
   30 PRINT'"THIS PROGRAM TESTS YOUR
TABLES SKILLS"
   40 INPUT '"WHICH NUMBER MULTIPLICATION
TABLE WOULD YOU LIKE TO BE TESTED ON
",number
   50 PRINT'"HERE GOES :-"
   60 PRINT
   70 FOR N=1 TO 10
   80 PRINT number;"*";N;"=";number *N
   90 FOR X=1 TO 1000: NEXT X
  100 NEXT N
  110 PRINT'"Study these carefully  before
going on."
  120 PRINT'"ARE YOU READY TO GO ON, IF
SO PRESS Y"
  130 INPUT ans$
  140 IF ans$="Y"THEN 150
  150 CLS
  160 PRINT'"Type in your answer as each
question    comes up."
  170 score=0:wrong=0
  180 FOR n=1 TO 10
  190 LET correct=number*n
  200 PRINT number;"*";n;"=";
  210 INPUT you
```

```
  220 IF you=correct THEN PROCright ELSE
PROCwrong
  230 NEXT n
  240 IF score=10 AND wrong=0 THEN PRINT
"Well done you got them all right.":ELSE
PROCagain
  250 FOR N=1 TO 255 STEP8
  260 SOUND 1,-15,N,1
  270 NEXT N
  280 PROCagain
  290 END
  300 DEF PROCwrong
  310 wrong=wrong+1
  320 PRINT:PRINT"No, wrong answer, add
";number; " to ";number*(n-1)
  330 PRINT :GOTO 200
  340 ENDPROC
  350 DEF PROCright
  360 PRINT "Correct":score=score+1
  370 ENDPROC
  380 DEF PROCagain
  390 PRINT' "Do you want another go,Y or
N ?"
  400 INPUT go$
  410 IF go$="Y" THEN RUN ELSE PRINT "GAME
OVER.":END
  420 ENDPROC
```

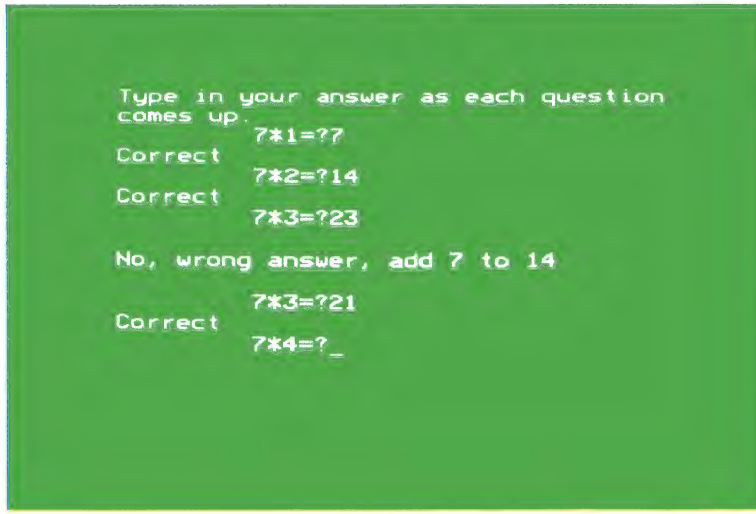
```
THIS PROGRAM TESTS YOUR TABLES SKILLS
WHICH NUMBER MULTIPLICATION TABLE WOULD
YOU LIKE TO BE TESTED ON ? 7

HERE GOES :-

          7*1=7
          7*2=14
          7*3=21
          7*4=28
          7*5=35
          7*6=42
          7*7=49
          7*8=56
          7*9=63
          7*10=70

Study these carefully before going on.

ARE YOU READY TO GO ON, IF SO PRESS Y
?
```



PROGRAM 25: LEO THE LION

This last program is a guessing game which uses X and Y co-ordinates. A 5 by 5 grid is drawn on the screen; you have to say which square you think Leo the lion is hiding in. You are allowed ten chances but you will need to remember which squares you've tried. This program tests your memory too !

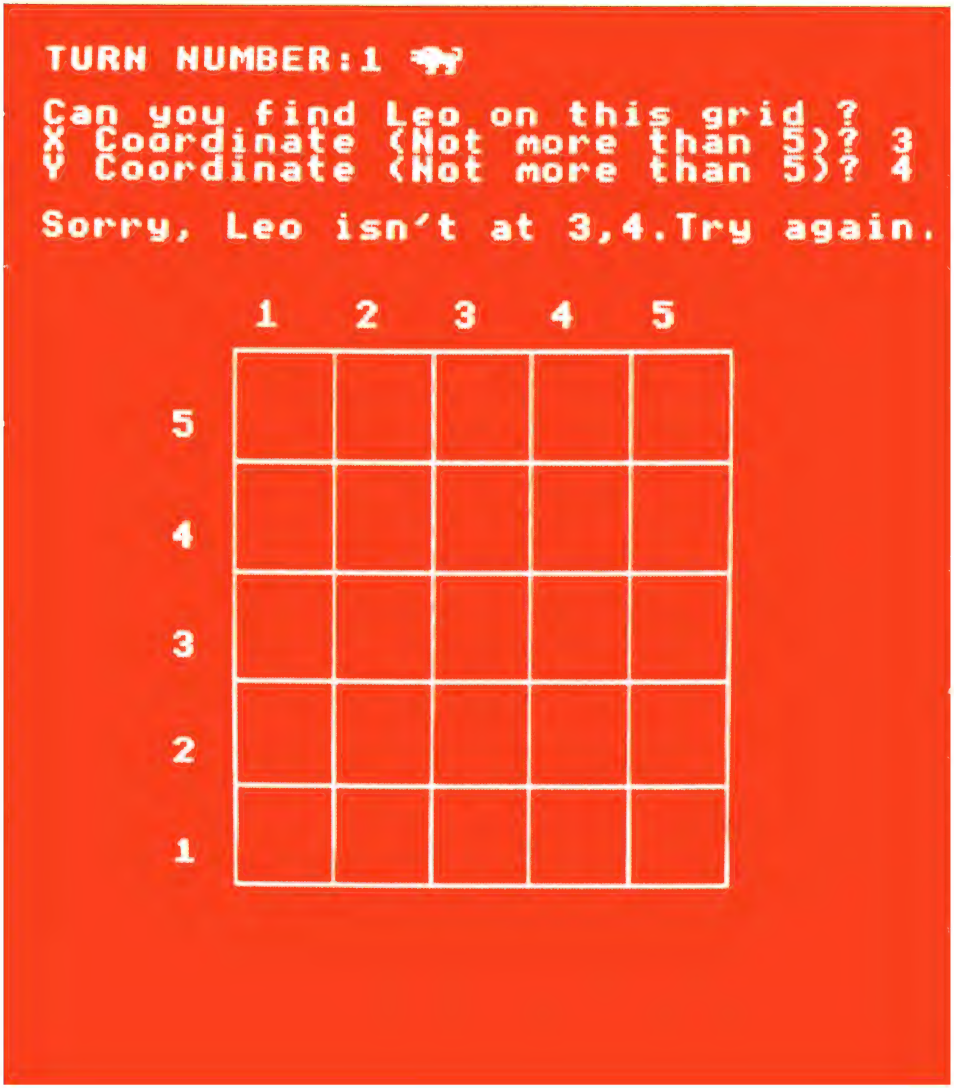
The program uses a lot of procedures. These are placed in the order of their use in the program at Lines 40 to 130.

PROClion (Lines 150 to 180) defines the graphics shapes which make up Leo.

PROCgrid (Lines 190 to 330) draws the grid on the screen and numbers the x and y axes.

PROChide (Line 350) decides where Leo is going to hide.

PROCguess (Lines 360 to 390) asks you to guess where Leo is hiding. This procedure uses a 'text window' at Line 380. This restricts part of the screen to text only. This enables the question section of the game to change without re-drawing the grid. If an incorrect guess is made then the program moves to PROCwrong. If the correct answer is given the program moves to PROCright.



PROCwrong (Lines 450 to 500) tell you that your guess was not correct and directs the program back to PROCguess. The instruction VDU 12 clears the 'text window' so that the next question can be asked.

PROCright (Lines 510 to 580) tell you that you've found Leo ! You are greeted with a sound effect to celebrate your success.

PROCquizend (Lines 660 to 680) tell you that you have not been successful and then direct the program to PROCshow.

PROCshow (Lines 590 to 650) show you where Leo was hiding and print the picture of him defined by PROClion.

With a little bit of imagination and thought you can make your own version of this program - perhaps even longer and better than this one. Have a go! The best way to learn programming is by trying out our own ideas. For example you could add a score, or more sound effects, or make the grid larger or smaller. This program has really only just begun !

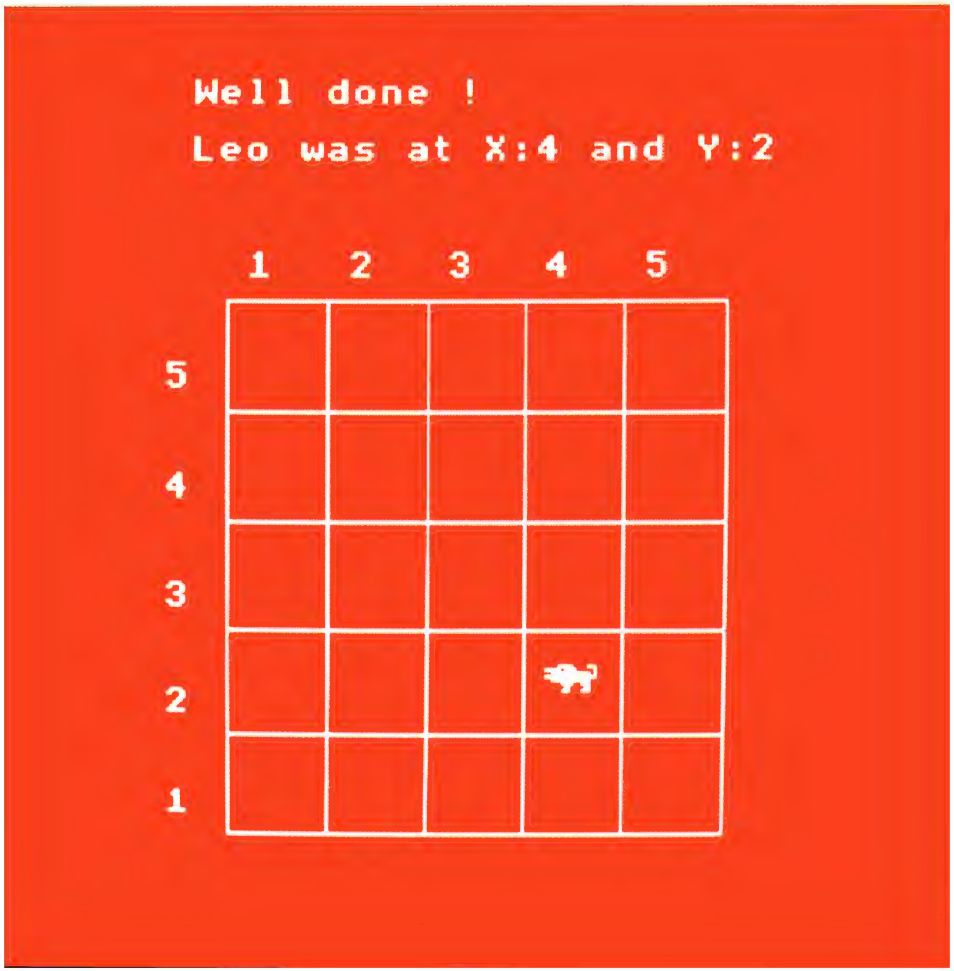
Type in:-

```
  10 REM * LIONSEARCH *
  20 MODE 4
  30 turn=1
  40 PROCgrid
  50 PROClion
  60 PROChide
  70 PROCguess
  80 PROCright
  90 PROCshow
 100 VDU 26:END
 110 PROCwrong
 120 PROCquizend
 130 PROCshow
 140 VDU 26:END
 150 DEF PROClion
 160 VDU 23,240,15,251,63,255,63,7,3,7
```

listing continues on next page

```
  170 VDU 23,241,135,193,253,255,252,156,12,28
  180 ENDPROC
  190 DEF PROCgrid
  200 FOR y=50 TO 650 STEP 120
  210 FOR x=300 TO 970 STEP 120
  220 MOVE x,y: DRAW 650,y
  230 MOVE x,y: DRAW x,650
  240 NEXT x
  250 NEXT y
  260 ref=1
  270 FOR x=325 TO 805 STEP 120
  280 MOVE x,700:VDU 5:PRINT;ref:ref=ref+1
  290 NEXT x
  300 ref=1
  310 FOR y=100 TO 580 STEP 120
  320 MOVE 225,y:VDU 5:PRINT;ref:ref=ref+1
  330 NEXT y
  340 ENDPROC
  350 DEF PROChide:X=RND(5):Y=RND(5):ENDPROC
  360 DEF PROCguess
  370 IF turn>10 THEN 120
  380 VDU 28,0,8,39,0:VDU 30:VDU4
  390 PRINT TAB(2,1);"TURN NUMBER:";turn;"
";CHR$(240);CHR$(241)
  400 PRINT TAB(2,3);"Can you find Leo on this
grid ?"
  410 INPUT TAB(2,4);"X Coordinate (Not more
than 5)",xcord
  420 INPUT TAB(2,5);"Y Coordinate (Not more
than 5)",ycord
  430 turn=turn+1:IF xcord=X AND ycord=Y THEN
80 ELSE 110
  440 ENDPROC
  450 DEF PROCwrong
  460 IF turn>10 THEN 120
  470 PRINT TAB(2,7);"Sorry, Leo isn't at
";xcord;",";ycord;".Try again."
  480 FOR N=1 TO 6000:NEXT
  490 VDU 12: PROCguess
  500 ENDPROC
  510 DEF PROCright
  520 VDU 12: PRINT TAB(8,6);"Well done !"
  530 FOR a=1 TO 5
  540 FOR N=1 TO 255 STEP 4
  550 SOUND 1,-15,N,.5
  560 NEXT N
```

```
  570 NEXT a
  580 ENDPROC
  590 DEF PROCshow
  600 rx=(X*120)+210:ry=(Y*120)+10
  620 MOVE rx,ry:VDU 5:PRINT
CHR$(240);CHR$(241)
  630 VDU 4: PRINT TAB(8,8);"Leo was at
X:";X;" and Y:";Y
  640 PRINT
  650 ENDPROC
  660 DEF PROCquizend
  670 VDU 12:PRINT TAB(8,4);"Sorry, you lose
!"
  680 ENDPROC
```



The programs you write can be 'saved' i.e. recorded, on cassette tape. These taped programs can be used over and over again.

Connect your computer to the cassette tape recorder, put a blank tape in and make sure that it is wound past the start of the tape.

Start by typing in a short program, just to make sure that you understand how to 'save' a program. Type in the program, press RETURN and then type in :-

```
SAVE "
```

You now have to give your program a name. Try to pick one that describes the program. For example Program 25 in this book could be called "LEO". The quotation marks must be used.

On your screen you should now see this:-

```
SAVE "LEO"
```

Press the RETURN key and you will see this message:-

RECORD then RETURN

Now press the RECORD and PLAY buttons on your cassette recorder, then press the RETURN key. You will now see your program's name appear on the screen, with a set of numbers.

When your computer has finished recording the program, it bleeps. You can now stop the tape. If your cassette recorder has a motor control, it will stop automatically.

Now that you have saved your program, you can load it back into your computer. First type in the command NEW, then press RETURN. This makes sure that your computer's memory is clear.

Rewind your tape back to the beginning, and type in the message:-

LOAD "LEO"

Press the RETURN key.

You will now see this message:-

Searching

Press the PLAY button on your cassette recorder and, after a few seconds, you will see the name of the program. This will be followed by some numbers. When the program has loaded, your computer bleeps.

Type in the word LIST and you should see your program listed.

If all goes well, you should have no problems with loading and saving. But problems do occur for all sorts of reasons. Your cassette recorder may not be

suitable for use with the Electron or BBC Microcomputer, you may not have the volume and tone controls set correctly, or you may not have the leads connected properly.

The USER GUIDE can tell you about loading and saving in more detail. If in doubt, ask your supplier or someone else who has a computer like yours. They will probably be able to help.

Above all, don't panic. Most micro users have problems from time to time!

SUMMARY

Symbols used

+	addition
-	subtraction
/	division
*	multiplication
^	raise to the power of
<	less than
>	greater than

Words and variables

RND: Selects a number at random from a given range. When followed by a number in brackets, the range is between 0 and that number.

MOD: Gives the remainder following a division sum.

DIV: Gives the whole number part produced by a division sum.

INT: Rounds down fractional numbers to the nearest whole number.

Variable names for numbers may be single letters or longer names; they may not begin with a number, nor may they contain any spaces or punctuation marks. Variable names for words ("strings") must be followed by a $ sign.

Modes

The BBC Micro has eight display modes; the seven display modes on the Electron are the same as the first seven of these (MODES 0-6).

MODES 3,6,and 7 can be used for text only. All the others can be used for graphics too.

MODES 2 and 5 offer the largest characters and the widest range of colours, but may not always be suitable for displays with lots of text. For these displays other modes are preferable. MODE 0 offers the highest resolution graphics, and 80 characters to a line.

Decimals & fractions

Decimal fractions containing more than 10 figures cannot be stored and will be rounded up where necessary.

Co-ordinates & tabs

To position text on the screen, you use the word TAB, followed by one or two numbers in brackets. One number fixes the start point of any text on a line. Where two numbers are used, the second number determines the line on the screen at which the text will be placed.

When designing graphics displays, the graphics command is followed by two co-ordinates. The first or X co-ordinate says where a point goes on the horizontal axis, the second or Y co-ordinate says where it goes on the vertical axis.

INDEX

Abbreviations 10
Addition 16, 17
Addition quiz 22–24, 30, 31
Addition series 47–49
AUTO 22

BASIC 5, 6, 10, 14
BREAK 12
Bugs 9

Capital letters 10, 11
CLS 12
Coordinates 63
COPY 13

Decimals 63
DELETE 7, 13
DIM 44
Direct Commands 19
DIV 18, 19, 62
Division 17, 18

EDIT 13
ESCAPE 22

FOR . . . NEXT loop 23
Fractions 63

INPUT 10, 15
INT 62

Lights 11
Line numbers 7, 22
Listing 8, 10, 12, 44
Loading 61, 62

MOD 18, 19, 62
Modes 63
Multiplication tables 52–54

NEW 10
Number crunchers 51

OLD 12

PI 42
PRINT 10
Programming 6–13
Punctuation 9

REM statement 17
Renumbering 34
REPEAT . . . UNTIL loop 21, 22
RETURN 7
RND 62
RUN 8

Saving 60
Spacing 9
Subtraction 50
Symbols 62
Syntax error 9

TAB 63

Variable names 63

PROGRAMS

Programs are listed under their REM statements.

Addition 16
Change Test 46
Circle Area 43
Conversion Prog 36
Different Routine 32
Division Using / 18
Double Height Display 30, 31
Lionsearch 55–59
Loop & Count 20
Miles to KM 38
MOD and DIV 19
Percentages 34
Priorities 41
REPEAT . . . UNTIL 22
Series Based on Addition 48, 49
Series Based on Subtraction 50
Shopping List 39, 40
Stars 26, 27
Tables Tester 52–54
Using * 15, 16
Village 29
Yes or No? 33

ACKNOWLEDGEMENTS

The author wishes to acknowledge his use of the 'Computer Concepts Graphics Rom' and the 'British Micro Grafpad' in the preparation of the graphics titles in this book. The VDU used for these and the screen shots of the programs was a 'Microvitec Cub Monitor'.

Have fun with your computer!

These programs – they start simple and become progressively more complex – will give you lots of fun and teach you how to experiment with and enjoy your micro. You can adapt the programs to suit your needs and by the end of the book you will be able to write your own programs.

Make friends with your micro!

TITLES INCLUDE:

SIMPLE SHAPES AND PICTURES
SIMPLE MUSIC AND SOUND EFFECTS
SIMPLE WORDS AND WORD GAMES
SIMPLE MATHS, TABLES AND NUMBERS
SIMPLE MOVEMENT AND ANIMATION
SIMPLE FACTS AND FIGURES

GRANADA PUBLISHING
0 246 12268 4

£1.95 net